

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	10	proxy same (complex with object) same generat\$	US-PGPUB; USPAT	OR	ON	2005/03/04 11:05
L2	16	proxy same (complex adj object)	US-PGPUB; USPAT	OR	ON	2005/03/04 12:00
L3	90	proxy same object same (pass\$ or parameter) same distributed	US-PGPUB; USPAT	OR	ON	2005/03/04 11:12
L4	18	proxy same object same (pass\$ or parameter) same distributed same generat\$	US-PGPUB; USPAT	OR	ON	2005/03/04 11:32
L5	8	proxy same object same (pass\$ or parameter) same generat\$ same (first with second)	US-PGPUB; USPAT	OR	ON	2005/03/04 11:20
L6	7	("4660142"   "5446901"   "5615362"   "5765174"   "5802367"   "5884316"   "5991823").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2005/03/04 11:29
L7	15	proxy same object same (pass\$ or parameter) same generat\$	EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/03/04 11:56
L8	28	proxy with pair	EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/03/04 11:56
L9	0	(proxy same (complex adj object)).clm.	US-PGPUB; USPAT	OR	ON	2005/03/04 12:01
L10	53	((complex adj object)).clm.	US-PGPUB; USPAT	OR	ON	2005/03/04 12:01
L11	0	((complex adj object)).clm. and proxy.ti,ab.	US-PGPUB; USPAT	OR	ON	2005/03/04 12:01
L12	4	((complex adj object)).clm. and proxy	US-PGPUB; USPAT	OR	ON	2005/03/04 12:01
S1	2657210	remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference	US-PGPUB; USPAT	OR	ON	2003/11/18 16:32
S2	29618	"709"/\$.ccls.	US-PGPUB; USPAT	OR	ON	2003/11/18 16:32
S3	36	"719"/\$.ccls.	US-PGPUB; USPAT	OR	ON	2003/11/18 16:32
S4	29641	"709"/\$.ccls. or "719"/\$.ccls.	US-PGPUB; USPAT	OR	ON	2003/11/18 16:32
S5	26684	(remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference) and ("709"/\$.ccls. or "719"/\$.ccls.)	US-PGPUB; USPAT	OR	ON	2003/11/18 16:32
S6	212696	remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference same (complex or composite or large)	US-PGPUB; USPAT	OR	ON	2003/11/18 16:33

S7	2520	((remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference) and ("709"/\$.ccls. or "719"/\$.ccls.)) and (remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference same (complex or composite or large))	US-PGPUB; USPAT	OR	ON	2003/11/18 16:33
S8	581	remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference same (complex or composite or large) same proxy	US-PGPUB; USPAT	OR	ON	2003/11/18 16:34
S9	309	((remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference) and ("709"/\$.ccls. or "719"/\$.ccls.)) and (remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference same (complex or composite or large))) and (remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference same (complex or composite or large) same proxy)	US-PGPUB; USPAT	OR	ON	2003/11/18 16:34
S10	464	remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference same (complex or composite or large) same proxy same (member adj function)	US-PGPUB; USPAT	OR	ON	2003/11/18 16:35
S11	242	((remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference) and ("709"/\$.ccls. or "719"/\$.ccls.)) and (remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference same (complex or composite or large))) and (remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference same (complex or composite or large) same proxy)) and (remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference same (complex or composite or large) same proxy same (member adj function))	US-PGPUB; USPAT	OR	ON	2003/11/18 16:35

S12	0	remote\$ same (call\$ or invocation or invok\$) same object same (parameter or reference) same (complex or composite or large) same proxy same (member adj function)	US-PGPUB; USPAT	OR	ON	2003/11/18 16:36
S13	7	remote\$ same (call\$ or invocation or invok\$) same object same (parameter or reference) same (complex or composite or large) same proxy	US-PGPUB; USPAT	OR	ON	2003/11/18 16:44
S14	9	("5218713"   "5247676"   "5307490"   "5457797"   "5481721"   "5822585"   "5960087"   "6157960"   "6173327").PN.	USPAT	OR	ON	2003/11/18 16:37
S15	7	("5421016"   "5632034"   "5898834"   "6223340"   "6272674"   "6295643"   "6360361").PN.	USPAT	OR	ON	2003/11/18 16:38
S16	1	"5566302".PN.	USPAT	OR	ON	2003/11/18 16:42
S17	104	proxy with Y	US-PGPUB; USPAT	OR	ON	2003/11/18 16:44
S18	25	((("5862328") or ("5511197") or ("5655101") or ("5687370") or ("5724588") or ("5822521") or ("5860072") or ("6026415") or ("6275937") or ("6618737") or ("5276901") or ("5903271") or ("6094657") or ("6182155") or ("6263485") or ("6304884") or ("6338117") or ("6385661") or ("6438576") or ("6457065") or ("6549955") or ("6567861") or ("6622175") or ("6018805") or ("6049838")).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2003/11/19 10:28
S19	25	((("6157960") or ("6366285") or ("5924116") or ("6012126") or ("6023270") or ("6065058") or ("6148438") or ("6230160") or ("6266742") or ("6324543") or ("6405262") or ("6425057") or ("6463508") or ("6487714") or ("6571262") or ("6598121") or ("6601090") or ("6601143") or ("6629112") or ("6629128") or ("5671345") or ("5761511") or ("5995975") or ("5995998") or ("6085035")).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2003/11/19 15:49
S20	8	((("6157960") or ("6230160") or ("6629128") or ("5511197") or ("5862328") or ("6049838") or ("6622175") or ("6549955")).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2003/11/19 16:13

S21	6	("5737607"   "5897634"   "5903725"   "5928335"   "5983233"   "6453333"   "2001/0003824").PN.	USPAT	OR	ON	2003/11/19 16:01
S22	1	("5838906").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2003/11/20 08:58
S23	11	("5793965"   "6012067"   "6151639"   "6182154"   "6182155"   "6230160"   "6253253"   "6260078"   "6347342"   "6385661"   "6405246").PN.	USPAT	OR	ON	2003/11/19 16:17
S24	5	("5457797"   "5995753"   "6009432"   "6044217"   "6049819").PN.	USPAT	OR	ON	2003/11/19 16:25
S25	7	("5369766"   "5613148"   "5642511"   "5655101"   "5732270"   "5862328"   "5903725").PN.	USPAT	OR	ON	2003/11/19 16:36
S26	3	("5329619"   "5442749"   "5452459").PN.	USPAT	OR	ON	2003/11/19 16:36
S27	9	("5511197"   "5577251"   "5781633"   "5793965"   "6157960"   "6182153"   "6182154"   "6182155"   "6230160").PN.	USPAT	OR	ON	2003/11/19 17:04
S28	8	("5136716"   "5481721"   "5640564"   "5684955"   "5768532"   "5822521"   "5822585"   "5948072").PN.	USPAT	OR	ON	2003/11/19 17:07
S29	1	(first adj proxy) same (second adj proxy) same object same (translat\$ or transfer\$)	US-PGPUB; USPAT	OR	ON	2003/11/20 09:23
S30	5	("5613148"   "5644766"   "5862325"   "5881230"   "5903725").PN.	USPAT	OR	ON	2003/11/20 09:07
S31	10	(first adj proxy) same (second adj proxy) same object	US-PGPUB; USPAT	OR	ON	2003/11/20 09:24
S32	1	("6011918").PN.	US-PGPUB; USPAT	OR	OFF	2005/03/03 15:59
S33	8	("6011918").URPN.	USPAT	OR	ON	2005/03/03 16:32
S34	5	("5485616"   "5535394"   "5581758"   "5892917"   "5920725").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2005/03/03 16:36
S35	26	("5485616").URPN.	USPAT	OR	ON	2005/03/03 16:51
S36	14	(first adj proxy) same (second adj proxy) same object	US-PGPUB; USPAT	OR	ON	2005/03/03 16:55
S37	3134451	remote\$ same (call\$ or invocation or invok\$) same object same parameter or reference	US-PGPUB; USPAT	OR	ON	2005/03/03 16:55

S38	2	(first adj proxy) same (second adj proxy) same object same (translat\$ or transfer\$)	US-PGPUB; USPAT	OR	ON	2005/03/03 16:56
S39	0	proxy same genrat\$3 same (complex adj object)	US-PGPUB; USPAT	OR	ON	2005/03/03 16:57
S40	3	proxy same generat\$3 same (complex adj object)	US-PGPUB; USPAT	OR	ON	2005/03/03 16:58
S41	66297	"719"/\$.ccls. or "709"/\$.ccls. or "717"/\$.ccls. or "707"/\$.ccls.	US-PGPUB; USPAT	OR	ON	2005/03/03 16:59
S42	86	(proxy and distribut\$3 and (object or program)).ti,ab.	US-PGPUB; USPAT	OR	ON	2005/03/03 17:01
S43	72	S41 and S42	US-PGPUB; USPAT	OR	ON	2005/03/03 17:00
S44	54144	(partition\$3 or split\$4).ti,ab.	US-PGPUB; USPAT	OR	ON	2005/03/03 17:00
S45	0	S43 and S44	US-PGPUB; USPAT	OR	ON	2005/03/03 17:00
S46	340	719/330.ccls.	US-PGPUB; USPAT	OR	ON	2005/03/03 17:01
S47	7	S42 and S46	US-PGPUB; USPAT	OR	ON	2005/03/04 11:01

**TDB-ACC-NO: NN980189**

**DISCLOSURE TITLE: Passing Proxies as Parameters to Methods and Return Values from Methods**

**PUBLICATION-DATA: IBM Technical Disclosure Bulletin, January 1998, US**

**VOLUME NUMBER: 41**

**ISSUE NUMBER: 1**

**PAGE NUMBER: 89 - 92**

**PUBLICATION-DATE: January 1, 1998 (19980101)**

**CROSS REFERENCE: 0018-8689-41-1-89**

**DISCLOSURE TEXT:**

Disclosed is a method for the use of proxy objects to enable transparent conversion of single-machine-based applications to a client-server framework. This scheme creates proxies for remote objects, which convert local method calls into Remote Method Invocation (RMI) calls. These RMI calls pass parameters by serializing them and passing the serialized data between machines. Return values from these method calls are also serialized and sent over the network. A class which is proxied in this way is considered split.

With this setup, a number of issues are brought up:

- o If a machine A is set up with FooBar as a proxy class instead of a real object, what happens when you want to send that machine an instance of the real FooBar? Simply sending it will not work, since your version of the FooBar class and A's version are different.
- o Objects passed between machines are passed by copy. Maintaining single-copy semantics is time-consuming when there are multiple copies of an object.

This invention disclosure proposes the generation of proxies for objects and sending those proxies across the network instead of sending the actual objects.

Unlike the real object, a proxy can be sent to a machine using that class as a proxy class without any confusion. Using a proxy

**class**

means that the data for the class is only stored in one place, giving you single-copy semantics for free.

For example, consider an application which performs matrix manipulations and displays the results. A user on a thin client may want to farm out the calculation to a more powerful server. In order to do this, simply set the home of the Matrix class to the server.

In

this case, there are many operations that may be implemented by passing

Matrix objects into a Matrix method, such as `mat3 = mat1.add(mat2)`.

In

this case, the proxies must take care of two operations:

1. Replacing the client version of `mat2` (a proxy) with the server version (the actual `mat2`)

- 2.

- Replacing the server version of `mat3` (the actual `mat3`) with the client version (a proxy)

This is what pseudocode for these routines in the proxies might look like:

**Server Proxy**

```
class MatrixProxy implements MatrixInterface {
    Matrix myObject;
    ...
    MatrixInterface add(MatrixInterface m2) {
        Matrix returnVal = myObject.add(m2.myObject);
        return new MatrixProxy built out of returnVal;
    }
}
```

**Client Proxy**

```
class Matrix {
    MatrixInterface serverProxy;
    ...
    Matrix add(Matrix m2) {
        MatrixInterface returnVal =
serverProxy.add(m2.serverProxy);
        return new client serverProxy built out of returnVal;
    }
}
```

In this setup, the conversion process from a proxy to a real object and back is done in two steps. This is done to avoid the naming

conflicts of having two different classes with the same name on the same machine.

#### **Proxy Generation**

To make this scheme useful, these proxies must be generated automatically when needed. This is possible with some modifications to the proxy generation code as follows.

First, assume that each server-side proxy makes available the object it proxies through a field or method call.

For simplicity, assume a field called `myObject` is used. Also, assume that the client-side proxy puts a stub of the server proxy in a field called `serverProxy`. Consider a pair of proxies as shown below:

##### **Server proxy**

```
class FooBarProxy implements FooBarInterface [  
    FooBar myObject;  
    FooBarProxy(FooBar arg) [myObject = arg;]  
    ...  
    Result fnord(Request arg1) [  
        return myObject.fnord(arg1);  
    ]  
    ...
```

##### **Client proxy**

```
class FooBar [  
    FooBarInterface serverProxy;  
    FooBar(FooBarInterface arg) [serverProxy = arg;]  
    ...  
    Result fnord(Request command) [  
        return serverProxy.fnord(command);  
    ]  
]
```

It would be desirable to modify the `FooBar` proxy pair to deal with `Request` and `Result` being proxies instead of the real classes.

#### **Server Proxy Modification**

To fix each method of the server-proxy:

1. Find all the arguments and return-values that are proxied to the client-side. Replace each with server proxy interfaces.
2. When these arguments are referenced in the body of the code, replace all instances of these references with references to the `myObject` field.
3. Instead of returning the result of the real objects method call directly, enclose it inside of a server proxy.

**Therefore, the fnord method of the server proxy becomes:**

```
ResultInterface fnord(RequestInterface arg1) [  
    return new ResultProxy(myObject.fnord(arg1));  
]
```

#### **Client Proxy Generation**

**Fixing the methods of the client proxy is done in a similar way:**

- 1. Fix the RMI call to the server so it conforms to the new server interface, by replacing all references of proxy objects with a reference to the server proxy interfaces they use.**

**2.**

**Instead of returning the server proxy interface (which is now done because of the change in the server proxy), build a client proxy out of the server proxy interface and return this instead.**

**Therefore, the method call becomes:**

```
Result fnord(Request arg1) [  
    return new Result(serverProxy.fnord(arg1.serverProxy));  
]
```

**SECURITY:** Use, copying and distribution of this data is subject to the restrictions in the Agreement For IBM TDB Database and Related Computer Databases. Unpublished - all rights reserved under the Copyright Laws of the United States. Contains confidential commercial information of IBM exempt from FOIA disclosure per 5 U.S.C. 552(b)(4) and protected under the Trade Secrets Act, 18 U.S.C. 1905.

**COPYRIGHT STATEMENT:** The text of this article is Copyrighted (c) IBM Corporation 1998. All rights reserved.